

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 11-203149

(43)Date of publication of application : 30.07.1999

(51)Int.Cl. G06F 9/46 -

(21)Application number : 10-004546

(71)Applicant : MATSUSHITA ELECTRIC IND CO LTD

(22)Date of filing : 13.01.1998

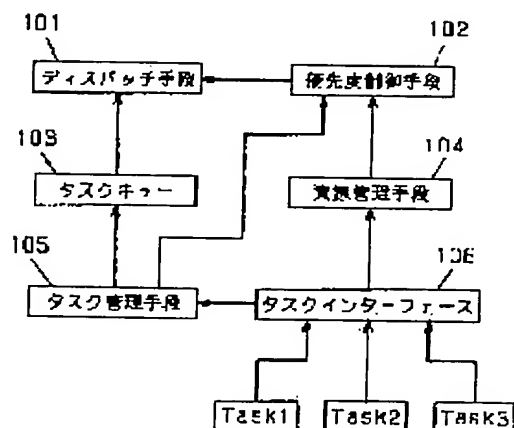
(72)Inventor : ZAIKI KOUJI

(54) DEVICE AND METHOD FOR SCHEDULING TASK

(57)Abstract:

PROBLEM TO BE SOLVED: To provide a task scheduling device which has a little overhead and prevents priority inversion from taking place.

SOLUTION: This device is provided with a task interface part 106 which receives a request from a task, a resource managing means 104 which manages resource information that is used by the task, a task managing means 105 which manages information about tasks, a priority controlling means 102 which decides the execution priority of a task with instructions from the means 104 and 105 and a dispatching part 101 which takes out and starts a task that has high priority from a task queue according to priority that is decided by the means 102, manages priority about a task which preliminarily executes the same exclusive executing part (shared resources) in each exclusive resource and when the exclusive executing part is carried out, it always makes priority high not so as to be interrupted by other tasks which have the possibility of executing the same exclusive executing part and carries out.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's

decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2003 Japan Patent Office

(19)日本国特許庁 (J P)

(12) 公 開 特 許 公 報 (A)

(11)特許出願公開番号

特開平11-203149

(43)公開日 平成11年(1999) 7月30日

(51)Int.Cl.⁹

G 0 6 F 9/46

識別記号

3 4 0

F I

G 0 6 F 9/46

3 4 0 B

3 4 0 F

審査請求 未請求 請求項の数5 O L (全 6 頁)

(21)出願番号 特願平10-4546

(22)出願日 平成10年(1998) 1月13日

(71)出願人 000005821

松下電器産業株式会社

大阪府門真市大字門真1006番地

(72)発明者 材木 幸治

大阪府門真市大字門真1006番地 松下電器

産業株式会社内

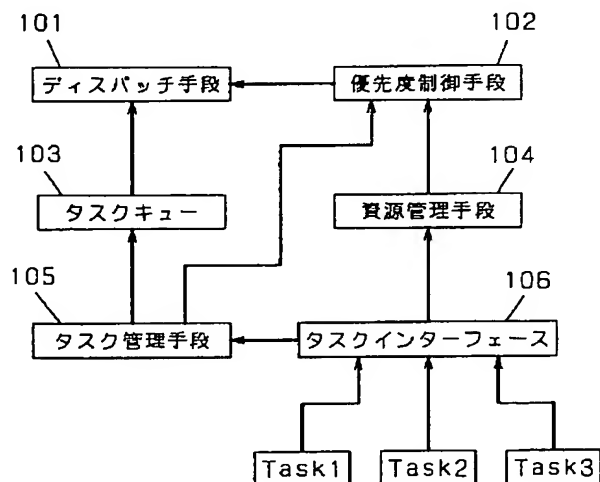
(74)代理人 弁理士 滝本 智之 (外1名)

(54)【発明の名称】 タスクスケジュール装置およびその方法

(57)【要約】

【課題】 少ないオーバーヘッドで優先度の逆転が発生しないタスクスケジュール装置を提供する。

【解決手段】 タスクからの要求を受け付けるタスクインターフェース部106と、タスクが使用する資源情報を管理する資源管理手段104と、タスクに関する情報を管理するタスク管理手段105と、資源管理手段104とタスク管理手段105からの指示でタスクの実行優先度を決定する優先度制御手段102と、優先度制御手段102によって決定された優先度に従って高い優先度のタスクをタスクキューから取り出して起動するディスパッチ部101とを備え、予め同じ排他的実行部(共有資源)を実行するタスクに関して優先度をその排他的資源ごとに管理して排他的実行部を実行するときは必ず、同じ排他的実行部を実行する可能性のある他のタスクに割り込まれないように優先度を高くして実行する。



【特許請求の範囲】

【請求項 1】 マルチタスク処理を行なうためのタスクスケジュール装置において、タスクからの要求を受け付けるタスクインターフェース部と、このタスクインターフェース部から与えられるタスクが使用する資源情報を管理する資源管理手段と、タスクインターフェース部から与えられるタスクに関する情報を管理するタスク管理手段と、このタスク管理手段からのタスク情報をもとに実行順を保持するタスクキューと、前記資源管理手段と前記タスク管理手段からの指示で共有資源を使用するタスクの優先度のうちのいずれか 1 つの優先度をタスクの実行優先度とする優先度制御手段と、この優先度制御手段によって決定された優先度に従って高い優先度のタスクから前記タスクキューから取り出して起動するディスパッチ部とを備えたことを特徴とするタスクスケジュール装置。

【請求項 2】 前記資源管理手段は、共有資源の識別子とその共有資源を使用するタスクの識別子とタスクの優先度を記憶する手段を備えるとともに、タスクインターフェース部からの指示により前記共有資源の識別子とその共有資源を使用するタスクの識別子と優先度を関連付けて記憶し、また前記タスクインターフェース部からの指示により前記共有資源の識別子とその共有資源を使用するタスクの識別子と優先度を記憶する手段から消去し、さらに優先度制御手段に前記共有資源に関連付けられたタスクのうち与えられた順位の優先度を与えることを特徴とする請求項 1 記載のタスクスケジュール装置。

【請求項 3】 前記優先度制御手段は、タスク管理手段から共有資源の識別子とその共有資源を使用するタスクの識別子を与えられ、これにより資源管理手段から前記共有資源の識別子に関連付けて記憶されたタスクのうち与えられた順位の優先度を取り出して、前記タスクの優先度を一時的にこの優先度に設定してディスパッチ部に通知することを特徴とする請求項 2 記載のタスクスケジュール装置。

【請求項 4】 マルチタスク処理を行なうためのタスクスケジュール方法において、タスクからの要求を受け付けるタスクインターフェース処理手続きと、このタスクインターフェース処理手続きから与えられるタスクが使用する資源情報を管理する資源管理手続きと、タスクインターフェース処理手続きから与えられるタスクに関する情報を管理するタスク管理手続きと、前記資源管理手続きと前記タスク管理手続きからの指示で共有資源を使用するタスクの優先度のうちのいずれか 1 つの優先度をタスクの実行優先度とする優先度制御手続きと、この優先度制御手続きによって決定された優先度に従って高い優先度のタスクから起動するディスパッチ手続きとを備えたことを特徴とするタスクスケジュール方法。

【請求項 5】 タスクからの共有資源要求がタスクインターフェース処理手続きからタスク管理手続きに渡さ

れ、前記タスク管理手続きから受けたタスクの優先度と既にこれと同じ共有資源を使用中のタスクの優先度との間の中間の優先度のタスクの実行が予め予測されたときに、優先度逆転の発生が起これと判断することを特徴とする請求項 4 記載のタスクスケジュール方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 本発明は、複数のタスクを並行して実行させるシステムにおいて、タスクの優先度逆転の検出およびそれを回避するタスクスケジューリング装置とそのタスクスケジューリング方法に関する。

【0002】

【従来の技術】 図 4 は複数のタスクで排他的に処理する部分を含む場合に、実行しなければならない排他制御用の命令を示したプログラム例である。図 4 において、402 は排他的に処理されなければならない部分を示しており、その前後に資源使用要求命令 404 と資源開放要求命令 405 を実行する。401、403 は排他的に実行しないタスクの処理部分を示している。このような排他的に実行する部分を含んだ複数のタスクが実行する動作を図 5、図 6 を用いて説明する。

【0003】 図 5 は第 1 の従来例のタスクスケジューリングの動作説明図である。3 つのタスク、Task1、Task2、Task3 が並行して実行する場合を考える。実行優先順位は高い方から、Task1、Task2、Task3 の順番であるとし、Task1 と Task3 が同じ資源を使用する、すなわち、排他的に実行されるものとする。Task3 が排他的な部分を実行中に、優先度の高い Task2 が発生 (501)。さらに優先度の高い Task1 が発生 (502)。ここで Task1 が排他的に実行すべき部分を実行しようとするすると Task3 が既に排他的な部分を実行中 (排他的な資源を使用中) なので、Task1 はロックされた状態で待機する。そこで Task2 へ制御が移る (503)。Task2 の処理が完了して Task3 が実行を再開し (504)、Task3 が排他的実行部分の処理を終えて資源開放を行なって Task1 の実行が再開する (505)。Task1 の処理が完了し、Task3 の実行が再開する (506)。

【0004】 図 6 は第 2 の従来例のタスクスケジューリングの動作説明図である。図 5 と同様に、3 つのタスク、Task1、Task2、Task3 が並行して実行する場合を考え、実行優先順位は高い方から、Task1、Task2、Task3 の順番であるとし、Task1 と Task3 が同じ資源を使用する、すなわち、排他的に実行されるものとする。Task3 が排他的な部分を実行中に、優先度の高い Task2 が発生 (601)。さらに優先度の高い Task1 が発生 (602)。ここで Task3 が排他的な部分を実行中 (共有資源を使用中) なので、Task1 はロックされた状態で待機する。ここで、Task3 の優先度を Task1 と同じ値に設定され、Task3 が排他的な部分を実行し (603)、排他的実行部分の処理を終えて資源開放を行なって Task1 の実

行が再開する（604）。Task1の処理が完了し、Task2へ制御が移る（605）。Task2の処理が完了してTask3が実行を再開する（606）。

【0005】

【発明が解決しようとする課題】しかしながら上記の第1の従来技術によれば、優先順位の低いタスク（Task3）が先に排他的に実行する部分を実行中に優先順位の高いタスク（Task1）が同じく排他的実行部分を実行しようとする、このタスク（Task1）はロックされた状態になる。この時、優先順位の中間のタスク（Task2）が実行を開始するとこれが終了するまでTask3の実行は待たされ、従って優先順位が一番高いタスク（Task1）の実行が待たされ、結局、優先順位の低いタスク（Task2）が優先順位の高いタスク（Task1）よりも先に終了してしまうという優先度の逆転が起きるという問題点があった。さらに第2の従来技術では、これを回避するため、一旦、優先度の高いタスク（Task1）が排他的実行部分を実行しようとしてロックされた時点で、優先順位の低いタスク（Task3）の優先度を上げて、排他的実行部分を実行させるようにするため、タスクのスイッチング回数が増え、オーバーヘッドが大きくなるという問題点があった。

【0006】本発明は上記問題点に鑑み、少ないオーバーヘッドで優先度の逆転が発生しないタスクスケジュール装置を提供することを目的とする。

【0007】また、優先度の高いタスク（Task1）が排他的実行部分を実行する場合は既に同じ排他的実行部分を実行している他のタスクが存在すればそこでロックされ、次に実行可能なタスクが実行されるようになってい、優先度の逆転が発生するかどうか判断できない。さらに、優先度の高いタスク（Task1）が排他的実行部分を実行しようとしてロックされた時点で、優先順位の低いタスク（Task3）の優先度を上げて、排他的実行部分を実行させるようにするため、必ずしも、優先度の逆転が起こらない場合でもこれを判定することはできないという問題点があった。

【0008】本発明は上記問題点に鑑み、確実に優先度の逆転を検出できるタスクスケジュール方法を提供することを目的とする。

【0009】

【課題を解決するための手段】上記課題を解決するため請求項1記載の本発明のタスクスケジュール装置は、タスクからの要求を受け付けるタスクインターフェース部と、タスクインターフェース部から与えられるタスクが使用する資源情報を管理する資源管理手段と、タスクインターフェース部から与えられるタスクに関する情報を管理するタスク管理手段と、タスク管理手段からのタスク情報をもとに実行順を保持するタスクキューと、前記資源管理手段とタスク管理手段からの指示でタスクの実行優先度を決定する優先度制御手段と、優先度制御手段

によって決定された優先度に従って高い優先度のタスクから前記タスクキューから取り出して起動するディスパッチ部とを備えたものであり、優先度制御手段は、タスク管理手段から共有資源を使用するタスクの識別子と共有資源の識別子を与えられ、これにより資源管理手段から前記共有資源の識別子に関連付けて記憶されたタスクのうち与えられた順位の優先度を取り出して、前記タスクの優先度を一時的にこの優先度に設定してディスパッチ部に通知することを特徴とするものである。

10 【0010】これにより、予め同じ排他的実行部（共有資源）を実行するタスクに関して優先度をその排他的資源ごとに管理して排他的実行部を実行するときは必ず、同じ排他的実行部を実行する可能性のある他のタスクに割り込まれないように優先度を高くして実行する。

20 【0011】また請求項5記載の本発明のタスクスケジュール方法は、タスクからの要求を受け付けるタスクインターフェース処理手続きと、タスクインターフェース処理手続きから与えられるタスクが使用する資源情報を管理する資源管理手続きと、タスクインターフェース処理手続きから与えられるタスクに関する情報を管理するタスク管理手続きと、前記資源管理手続きとタスク管理手続きからの指示でタスクの実行優先度を決定する優先度制御手続きと、優先度制御手続きによって決定された優先度に従って高い優先度のタスクから起動するディスパッチ手続きとを備え、タスクからの共有資源要求がタスクインターフェース処理手続きからタスク管理手続きに渡され、前記タスク管理手続きから受けたタスクの優先度と既にこれと同じ共有資源を使用中のタスクの優先度との間の中間の優先度のタスクの実行が予め予測されれば、優先度逆転の発生が起こると判断することを特徴とするものである。

30 【0012】これにより、優先度の高いタスク（Task1）が排他的実行部分を実行しようとしてロックされた時点で、このタスクの優先度と既に排他的実行部分を実行しているタスク（Task3）の優先度の間の優先度を持つタスクが存在するかで確実に優先度の逆転を検出する。

【0013】

40 【発明の実施の形態】本発明のタスクスケジュール装置及びそのタスクスケジュール方法の実施の形態について具体例を挙げて説明する。

50 【0014】図1は本発明の実施の形態におけるタスクスケジュール装置の構成を示す図である。図において、タスクインターフェース106は、Task1、Task2、Task3からの資源制御コマンドを受け付けてそのコマンドによって資源識別子とタスク識別子およびそのタスクの優先度をタスク管理手段105と資源管理手段104に送る。資源管理手段104は、タスクインターフェース106から与えられた資源識別子とそれを使用するタスクの識別子およびそのタスクの優先度とを関連付けて記憶または記憶から削除を行ない、資源識別子とそれに関連

付けられたタスクの優先度を優先度制御手段102に送る。優先度制御手段102は、タスク管理手段105から与えられる資源識別子からこれに関連するタスクの優先度を資源管理手段104から受けてその優先度のうち与えられた順位の優先度をディスパッチ手段101へ送る。タスクキュー103はタスク管理手段105から与えられるタスク識別子をその優先度と共に保持してディスパッチ手段101へ送る。ディスパッチ手段101は、優先度制御手段102からのタスク識別子とその優先度と、タスクキュー103からの最高優先度のタスク識別子とその優先度を受取り、これらを比較して優先度の大きい方のタスクの起動を行なう。

【0015】以上のように構成された本発明の実施の形態におけるタスクスケジューリング装置について、図1、図2、図4、図7を用いてその動作を説明する。

【0016】図2は本発明のタスクスケジューリング方法を示すフローチャート図である。資源制御コマンド解析手続き201でタスクからの資源制御コマンドから資源識別子を取り出し、その結果から資源制御コマンドの判定手続き202では資源の使用を登録/抹消または資源の使用を要求/開放を判定する。資源制御コマンドの判定手続き202で資源使用の登録/抹消と判定された場合は資源管理テーブルへの登録/抹消手続き203で資源識別子とタスク識別子および優先度を資源管理テーブルへ登録/抹消を行ない、次に優先度の決定手続き204を行なう。また、資源制御コマンドの判定手続き202で資源使用の要求/開放と判定された場合は次に優先度の決定手続き204を行なう。優先度の決定手続き204では、資源要求の場合はその資源を使用するタスクのうち与えられた順位の優先度のタスクの識別子とその優先度をディスパッチ処理手続き205へ渡す。資源開放の場合はその資源を使用したタスクの元の優先度とそのタスクの識別子をディスパッチ処理手続き205へ渡す。また、資源管理テーブルへの登録/抹消手続き203の後での優先度の決定手続き204では資源を登録/抹消要求したタスクの識別子とその優先度をディスパッチ処理手続き205へ渡す。ディスパッチ処理手続き205は、タスクキュー103と優先度の決定手続き204から渡されるタスクの優先度のうち大きい方の優先度のタスクを取り出して、そのタスクを起動する。

【0017】図7は本発明の実施の形態におけるタスクスケジューリングの動作説明図であり、これは図4に示すような排他的に実行する部分をもつタスクが実行する場合のスケジューリングを示している。3つのタスク、Task1、Task2、Task3を考える。各タスクの実行優先度をそれぞれ、Priority1、Priority2、Priority3とする。実行優先度は高い方から、Priority1、Priority2、Priority3の順番であるとし(Priority1>Priority2>Priority3)、Task1とTask3が同じ資源を使用する、すなわち、排他的に実行するものとする。予め3つのタス

クの資源使用がわかっているものとする、すなわち、Task1とTask3が同一資源(これをResource1とする)を使用し、Task2が別の資源(これをResource2とする)を使用するものとする。まずタスクの実行前に資源使用登録要求をタスクスケジューリング装置に発行することにより資源管理手段104は、資源管理テーブルへの登録/抹消手続き203によりタスクの資源使用情報を登録する。この時、Task1とTask3は同一資源(Resource1)を使用するものとして登録され、Task2は別の資源(Resource2)を使用するものとして登録される。

【0018】まず最初にTask3が排他的な部分を実行するためにタスクスケジューリング装置に対して資源使用要求コマンドを発行する。この時、例えばここではこの共有資源を使うタスクのうちの1番優先度の高いタスクの優先度で実行することを要求する。これにより、優先度制御手段102は、優先度決定手続き204により資源管理手段104から、登録されているTask3の使用する資源(Resource1)を使用するタスクのうち優先度の最も高いPriority1を獲得する。次にディスパッチ手段101によりTask3の実行が継続される。このとき、Task3は実行優先度 Priority1で実行することになる。Task3が排他的な部分を実行中に、Task2が発生したとすると、Task2はPriority2であり、Task3は本来、Priority3(Priority2>Priority3)であるが排他的な実行部分に関してはPriority1(Priority1>Priority2)で実行している。このため Task3は排他的な部分の実行が終了するまで継続して実行される。Task3が排他的実行部分を終了した時点で、タスクスケジューリング装置に対して資源開放要求コマンドを発行する。これにより、優先度制御手段102は、優先度の決定手続き204により資源管理手段104からTask3の本来の優先度Priority3を獲得する。次にディスパッチ手段101により、この時点で実行可能なタスクのうち実行優先度の高いTask2の実行が開始(701)される。

【0019】Task2の実行中にTask1が発生したとすると、実行優先度はTask1の方が高い(Priority1>Priority2)のでTask1の実行が開始(702)される。Task1の実行が終了すると、この時点で実行待ちの状態にある2つのタスクのうち実行優先度の高いTask2が実行を再開(703)し、Task2の実行が終了すると、Task1が実行を再開(704)する。

【0020】以上のように本発明の実施の形態によれば、予め同じ排他的実行部(共有資源)を実行するタスクに関して優先度をその排他的資源ごとに管理して排他的実行部を実行するときは必ず、同じ排他的実行部を実行する可能性のある他のタスクに割り込まれないように指定された順位の優先度を設定してタスクを実行する。

【0021】図3はタスクスケジューリングにおいてタスクの優先度逆転を検出する方法を示すフローチャート図である。

【0022】手続き301で、タスクから資源使用要求があった場合にすでにこれと同じ共有資源が使用中であるかどうかを判断する。使用中の場合はさらに手続き302で、共有資源を使用中のタスクの優先度と自タスクの優先度の間の優先度を持つタスクが存在するかどうかを判断する。存在すれば手続き304で優先度逆転の可能性ありと判断されその旨がタスク管理手段105に通知される。存在しなければ手続き305で優先度逆転の可能性無しと判断されその旨がタスク管理手段105に通知される。手続き301で同じ共有資源が使用中ではないと判断された場合は、手続き303で優先度制御手段102で決定された優先度に自タスクの優先度を一時的に変える。

【0023】以上のように本発明のタスクの優先度逆転の検出方法によれば、優先度の高いタスクが排他的実行部分を実行しようとしてロックされた時点で、このタスクの優先度と既に排他的実行部分を実行しているタスクの優先度の間の優先度を持つタスクが存在するかで確実に優先度の逆転を検出する。

【0024】

【発明の効果】以上のように本発明のタスクスケジューリング装置及びそのタスクスケジューリング方法によれば、複数のタスクで同一資源を使用する場合、その資源を使用する可能性のあるタスクの実行優先度のうちの指定された順位の優先順位で資源を使用する処理を実行するため、資源使用要求を処理する際に、既に別のタスクにより予期できない中断がなくなり、本来の実行優先順位が逆転してしまうことがなくなる。また、予めタスクが使用する資源情報を登録し排他的実行部を実行開始する時点でそのタスクの実行優先度を変更するのでタスクの切替に要するオーバーヘッドを削減できるという効果が得られる。

*

* 【0025】さらに、本発明のタスクの優先度逆転の検出方法によれば、優先度の高いタスクが排他的実行部分を実行しようとしてロックされた時点で、このタスクの優先度と既に排他的実行部分を実行しているタスクの優先度の間の優先度を持つタスクが存在するかで確実に優先度の逆転を検出するため、逆転が存在しない場合は無駄なオーバーヘッドを減らすことができタスクスイッチ処理効率の向上が図れるという効果が得られる。

【図面の簡単な説明】

【図1】本発明の実施の形態におけるタスクスケジューリング装置の構成図

【図2】本発明の実施の形態におけるタスクスケジューリング方法を示すフローチャート

【図3】タスクの優先度逆転を検出する方法を示すフローチャート

【図4】排他制御を必要とするプログラム例の一部を示す図

【図5】第1の従来例のタスクスケジューリングの動作説明図

20 【図6】第2の従来例のタスクスケジューリングの動作説明図

【図7】本発明の実施の形態におけるタスクスケジューリングの動作説明図

【符号の説明】

101 ディスパッチ手段

102 優先度制御手段

104 資源管理手段

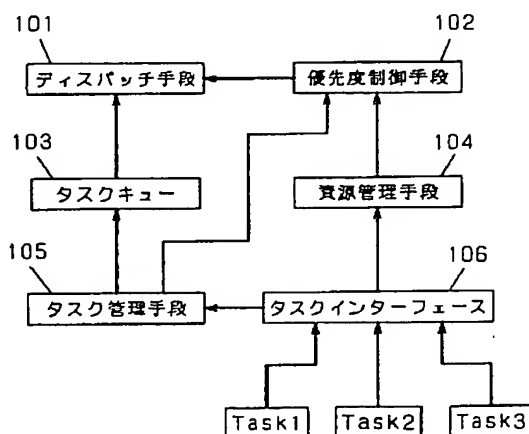
105 タスク管理手段

106 タスクインターフェース手段

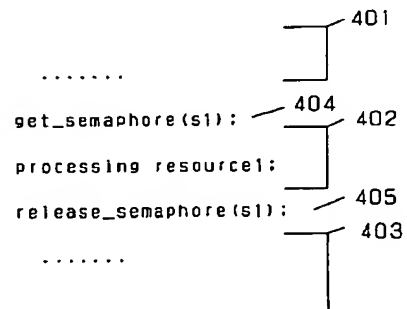
201 資源制御コマンド解析手続き

203 資源管理テーブルへの登録/抹消手続き

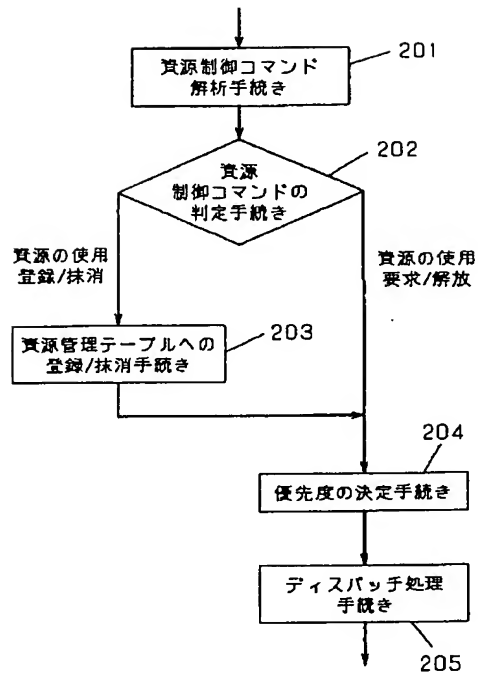
【図1】



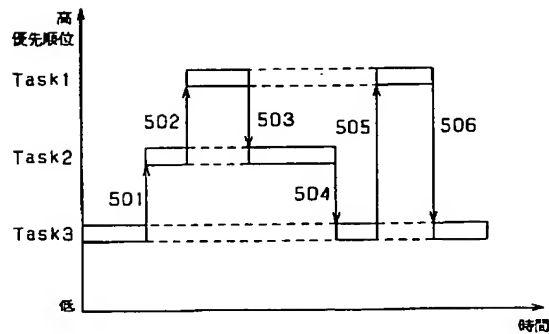
【図4】



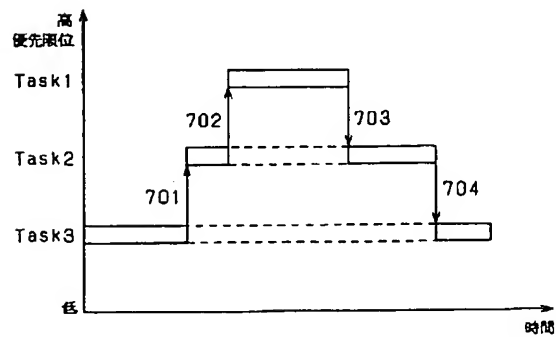
【図2】



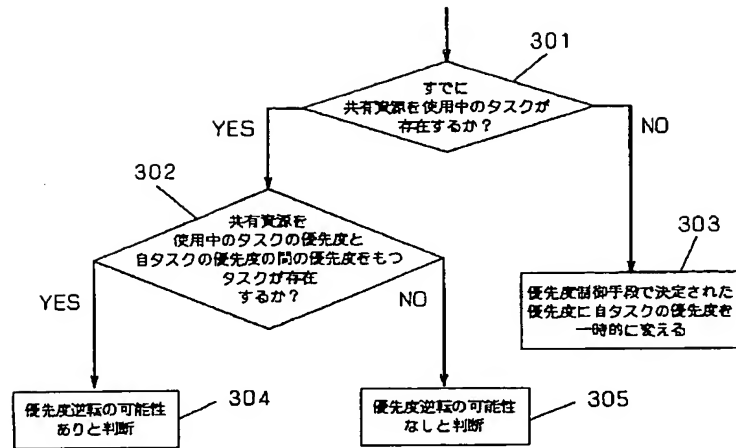
【図5】



【図7】



【図3】



【図6】

